IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A computer-implemented method for creating a graphical data flow program, wherein the graphical data flow program is ~~operable~~ configured to invoke a method of an object, wherein the method for creating the graphical data flow program operates in a computer including a display and a user input device, the method for creating the graphical data flow program comprising:

displaying on the screen a node in the graphical data flow program in response to user input, wherein the node is ~~operable~~ configured to invoke a method of an object, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

configuring the node to receive information on the object in response to user input, wherein said configuring comprises connecting the information on the object to an input of the node;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to invoke the method of the object during execution of the graphical data flow program.

2. (Original) The computer-implemented method of claim 1, wherein the node includes an object reference input for receiving a reference to the object;

wherein said configuring comprises connecting said object reference input of the node to receive the reference to the object;

wherein the node receives the information on the object on the object reference input during execution of the graphical data flow program.

3. (Original) The computer-implemented method of claim 2, wherein said configuring comprises:

displaying on the screen an object reference node which includes an object reference output that provides the reference to the object; and

connecting the object reference output of the object reference node to the object reference input of the node.

4. (Previously Presented) The computer-implemented method of claim 3, further comprising:

executing the graphical data flow program, wherein said executing includes propagating the reference to the object from the object reference output of the object reference node to the object reference input of the node.

5. (Original) The computer-implemented method of claim 1, further comprising:

executing the graphical data flow program, wherein said executing includes propagating the information on the object to the node.

6. (Previously Presented) The computer-implemented method of claim 1, wherein the object is comprised in a server, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input;

displaying on the screen a list of possible classes from the selected library;

selecting a class from the list of possible classes in response to user input;

wherein the object is instantiated from the class.

7. (Original) The computer-implemented method of claim 1, further comprising:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the method of the object; and

executing said execution instructions, wherein the node invokes the method of the object during said executing.

8. (Original) The computer-implemented method of claim 7, wherein said executing includes propagating the information on the object to the node.

9. (Currently Amended) The computer-implemented method of claim 1, wherein the graphical data flow program is ~~operable~~ configured to invoke the method of the object for performing instrumentation functions on an instrument.

10. (Original) The computer-implemented method of claim 1, wherein the graphical data flow program includes a block diagram and a front panel, wherein the block diagram includes the node.

11. (Currently Amended) A computer-implemented method for creating a graphical data flow program, wherein the graphical data flow program is ~~operable~~ configured to invoke a property of an object, wherein the method for creating the graphical data flow program operates in a computer including a display screen and a user input device, the method for creating the graphical data flow program comprising:

displaying on the screen a node in the graphical data flow program in response to user input, wherein the node is ~~operable~~ configured to invoke a property of an object, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

configuring the node to receive information on the object in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to invoke the ~~method~~ property of the object during execution of the graphical data flow program.

12. (Original) The computer-implemented method of claim 11, wherein the node includes an object reference input for receiving a reference to the object;

wherein said configuring comprises connecting the object reference input of the node to receive the reference to the object;

wherein the node receives the information on the object on the object reference input during execution of the graphical data flow program.

13. (Original) The computer-implemented method of claim 12, wherein said configuring comprises:

displaying on the screen an object reference node which includes an object reference output that provides the reference to the object; and

connecting the object reference output of the object reference node to the object reference input of the node.

14. (Previously Presented) The computer-implemented method of claim 13, further comprising:

executing the graphical data flow program, wherein said executing includes propagating the reference to the object from the object reference output of the object reference node to the object reference input of the node.

15. (Previously Presented) The computer-implemented method of claim 11, further comprising:

executing the graphical data flow program, wherein said executing includes providing the information on the object to the node.

16. (Original) The computer-implemented method of claim 11, wherein the object is comprised in a server, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input

displaying on the screen a list of possible classes from the selected library;

selecting a class from the list of possible classes in response to user input;

wherein the object is instantiated from the class.

17. (Original) The computer-implemented method of claim 11, further comprising:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the property of the object; and

executing said execution instructions, wherein the node invokes the property of the object during said executing.

18. (Currently Amended) The computer-implemented method of claim 11, wherein the node is ~~operable~~ configured to get and/or set one or more properties of the object.

19. (Currently Amended) The computer-implemented method of claim 11, wherein the graphical data flow program is ~~operable~~ configured to invoke the property of the object for performing instrumentation functions on an instrument.

20. (Original) The computer-implemented method of claim 11, wherein the graphical data flow program includes a block diagram and a front panel, wherein the block diagram includes the node.

21. (Currently Amended) A non-transitory memory medium comprising program instructions for creating a graphical data flow program, wherein the graphical data flow program is ~~operable~~ configured to invoke a method of an object, wherein the program instructions are executable to:

display on the screen a node in the graphical data flow program in response to user input, wherein the node is ~~operable~~ configured to invoke a method of an object, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

configure the node to receive information on the object in response to user input, wherein said configuring comprises connecting the information on the object to an input of the node;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to invoke the method of the object during execution of the graphical data flow program.

22. (Currently Amended) The non-transitory memory medium of claim 21, wherein the node includes an object reference input for receiving a reference to the object;

wherein said configuring comprises connecting the object reference input of the node to receive the reference to the object;

wherein the node receives the information on the object on the object reference input during execution of the graphical data flow program.

23. (Currently Amended) The non-transitory memory medium of claim 22, wherein said configuring comprises:

displaying on the screen an object reference node which includes an object reference output that provides the reference to the object; and

connecting the object reference output of the object reference node to the object reference input of the node.

24. (Currently Amended) The non-transitory memory medium of claim 23, wherein the program instructions are further executable to:

execute the graphical data flow program, wherein said executing includes propagating the reference to the object from the object reference output of the object reference node to the object reference input of the node

25. (Currently Amended) The non-transitory memory medium of claim 21, wherein the program instructions are further executable to:

construct execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the method of the object; and

execute said execution instructions, wherein the node invokes the method of the object during said executing.

26. (Currently Amended) The non-transitory memory medium of claim 25, wherein said executing includes propagating the information on the object to the node.

27. (Currently Amended) The non-transitory memory medium of claim 21, wherein the graphical data flow program is ~~operable~~ configured to invoke the method of the object for performing instrumentation functions on an instrument.

28. (Currently Amended) The non-transitory memory medium of claim 21, wherein the graphical data flow program includes a block diagram and a front panel, wherein the block diagram includes the node.

29. (Currently Amended) A non-transitory memory medium comprising program instructions for creating a graphical data flow program, wherein the graphical data flow program is ~~operable~~ configured to invoke a property of an object, wherein the program instructions are executable to:

display on the screen a node in the graphical data flow program in response to user input, wherein the node is ~~operable~~ configured to invoke a property of an object, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

configure the node to receive information on the object in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to invoke the ~~method~~ property of the object during execution of the graphical data flow program.

30. (Currently Amended) The non-transitory memory medium of claim 29, wherein the node includes an object reference input for receiving a reference to the object;

wherein said configuring comprises connecting the object reference input of the node to receive the reference to the object;

wherein the node receives the information on the object on the object reference input during execution of the graphical data flow program.

31. (Currently Amended) The non-transitory memory medium of claim 30, wherein said configuring comprises:

displaying on the screen an object reference node which includes an object reference output that provides the reference to the object; and

connecting the object reference output of the object reference node to the object reference input of the node.

32. (Currently Amended) The non-transitory memory medium of claim 31, wherein the program instructions are further executable to:

execute the graphical data flow program, wherein said executing includes propagating the reference to the object from the object reference output of the object reference node to the object reference input of the node.

33. (Currently Amended) The non-transitory memory medium of claim 29, wherein the program instructions are further executable to:

construct execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the property of the object; and execute said execution instructions, wherein the node invokes the property of the object during said executing.

34. (Currently Amended) The non-transitory memory medium of claim 29, wherein the node is ~~operable~~ configured to get and/or set one or more properties of the object.

35. (Currently Amended) The non-transitory memory medium of claim 29, wherein the graphical data flow program is ~~operable~~ configured to invoke the property of the object for performing instrumentation functions on an instrument.

36. (Currently Amended) The non-transitory memory medium of claim 29, wherein the graphical data flow program includes a block diagram and a front panel, wherein the block diagram includes the node.

37. (Currently Amended) A non-transitory memory medium which stores a graphical data flow program;

wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes, and wherein the graphical data flow program includes a node which is ~~operable~~ configured to invoke a method of an object;

wherein the node includes an input which is configurable to receive information on the object in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to invoke the method of the object during execution of the graphical data flow program.

38. (Currently Amended) A non-transitory memory medium which stores a graphical data flow program;

wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes, and wherein the graphical data flow program includes a node which is ~~operable~~ configured to invoke a property of an object;

wherein the node includes an input which is configurable to receive information on the object in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to invoke the ~~method~~ property of the object during execution of the graphical data flow program.

39. (Currently Amended) A computer-implemented method for creating a graphical data flow program, the method for creating the graphical data flow program comprising:

utilizing a computer to perform:

displaying on a display a graphical data flow program, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

displaying on the display a node in the graphical data flow program in response to user input, wherein the node is configurable to invoke a method of any of a plurality of software objects;

connecting the node to one or more other nodes in the graphical data flow program in response to user input;

configuring the node to receive information on a first object in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to invoke a method of the first object during execution of the graphical data flow program.

40. (Previously Presented) The computer-implemented method of claim 39,

wherein the lines interconnecting the plurality of nodes indicate that data produced by one node is used by another node.

41. (Previously Presented) The computer-implemented method of claim 39,

wherein said configuring comprises connecting the information on the first object to an input of the node.

42. (Previously Presented) The computer-implemented method of claim 39, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input;

displaying on the screen a list of possible classes from the selected library;

selecting a class from the list of possible classes in response to user input;

wherein the first object is instantiated from the class.

43. (Currently Amended) The computer-implemented method of claim 39, further comprising:

utilizing the computer to perform:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the method of the first object; and

executing said execution instructions, wherein the node invokes the method of the first object during said executing.

44. (Currently Amended) The computer-implemented method of claim 39, further comprising:

utilizing the computer to perform:

configuring the node to receive information on a first method of the first object in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to invoke the first method of the first object during execution of the graphical data flow program.

45. (Previously Presented) The computer-implemented method of claim 39,

wherein said configuring the node to receive information on a first object comprises configuring the node to receive information on a first class of the first object, wherein the first object is instantiated from the first class.

46. (Currently Amended) A non-transitory memory medium comprising program instructions for creating a graphical data flow program, wherein the program instructions are executable to implement:

displaying on a display a graphical data flow program, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

displaying on the display a node in the graphical data flow program in response to user input, wherein the node is configurable to invoke a method of any of a plurality of software objects;

connecting the node to one or more other nodes in the graphical data flow program in response to user input;

configuring the node to receive information on a first object in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to invoke a method of the first object during execution of the graphical data flow program.

47. (Currently Amended) The non-transitory memory medium of claim 46,

wherein the lines interconnecting the plurality of nodes indicate that data produced by one node is used by another node.

48. (Currently Amended) The non-transitory memory medium of claim 46,

wherein said configuring comprises connecting the information on the first object to an input of the node.

49. (Currently Amended) The non-transitory memory medium of claim 46, wherein said configuring comprises:

displaying on the screen a list of possible classes;

selecting a class from the list of possible classes in response to user input;

wherein the first object is instantiated from the class.

50. (Currently Amended) The non-transitory memory medium of claim 46, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input;

displaying on the screen a list of possible classes from the selected library;

selecting a class from the list of possible classes in response to user input;

wherein the first object is instantiated from the class.

51. (Currently Amended) The non-transitory memory medium of claim 46, wherein the program instructions are further executable to implement:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the method of the first object; and

executing said execution instructions, wherein the node invokes the method of the first object during said executing.

52. (Currently Amended) The non-transitory memory medium of claim 46, wherein the program instructions are further executable to implement:

configuring the node to receive information on a first method of the first object in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to invoke the first method of the first object during execution of the graphical data flow program.

53. (Currently Amended) The non-transitory memory medium of claim 46,

wherein said configuring the node to receive information on a first object comprises configuring the node to receive information on a first class of the first object, wherein the first object is instantiated from the first class.

54. (Currently Amended) A computer-implemented method for creating a graphical data flow program, the method for creating the graphical data flow program comprising:

utilizing a computer to perform:

displaying on a display a graphical data flow program, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

displaying on the display a node in the graphical data flow program in response to user input, wherein the node is configurable to perform at least one of getting or setting at least one property of any of a plurality of software objects;

connecting the node to one or more other nodes in the graphical data flow program in response to user input;

configuring the node to receive information on a first object in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to perform at least one of getting or setting at least one property of the first object during execution of the graphical data flow program.

55. (Previously Presented) The computer-implemented method of claim 54,

wherein the lines interconnecting the plurality of nodes indicate that data produced by one node is used by another node.

56. (Previously Presented) The computer-implemented method of claim 54,

wherein said configuring comprises connecting the information on the first object to an input of the node.

57. (Previously Presented) The computer-implemented method of claim 54, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input;

displaying on the screen a list of possible classes from the selected library;

selecting a class from the list of possible classes in response to user input;

wherein the first object is instantiated from the class.

58. (Currently Amended) The computer-implemented method of claim 54, further comprising:

utilizing the computer to perform:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to perform at least one of getting or setting at least one property of the first object; and

executing said execution instructions, wherein the node performs at least one of getting or setting at least one property of the first object during said executing.

59. (Currently Amended) The computer-implemented method of claim 54, further comprising:

utilizing the computer to perform:

configuring the node to receive information on a first property of the first object in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to perform at least one of getting or setting at least one property of the first object during execution of the graphical data flow program.

60. (Previously Presented) The computer-implemented method of claim 54,

wherein said configuring the node to receive information on a first object comprises configuring the node to receive information on a first class of the first object, wherein the first object is instantiated from the first class.

61. (Currently Amended) A non-transitory memory medium comprising program instructions for creating a graphical data flow program, wherein the program instructions are executable to implement:

displaying on a display a graphical data flow program, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

displaying on the display a node in the graphical data flow program in response to user input, wherein the node is configurable to perform at least one of getting or setting at least one property of any of a plurality of software objects;

connecting the node to one or more other nodes in the graphical data flow program in response to user input;

configuring the node to receive information on a first object in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to perform at least one of getting or setting at least one property of the first object during execution of the graphical data flow program.

62. (Currently Amended) The non-transitory memory medium of claim 61,

wherein the lines interconnecting the plurality of nodes indicate that data produced by one node is used by another node.

63. (Currently Amended) The non-transitory memory medium of claim 61,

wherein said configuring comprises connecting the information on the first object to an input of the node.

64. (Currently Amended) The non-transitory memory medium of claim 61, wherein said configuring comprises:

displaying on the screen a list of possible classes;

selecting a class from the list of possible classes in response to user input;

wherein the first object is instantiated from the class.

65. (Currently Amended) The non-transitory memory medium of claim 61, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input;

displaying on the screen a list of possible classes from the selected library;

selecting a class from the list of possible classes in response to user input;

wherein the first object is instantiated from the class.

66. (Currently Amended) The non-transitory memory medium of claim 61, wherein the program instructions are further executable to implement:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to perform at least one of getting or setting at least one property of the first object; and

executing said execution instructions, wherein the node performs at least one of getting or setting at least one property of the first object during said executing.

67. (Currently Amended) The non-transitory memory medium of claim 61, wherein the program instructions are further executable to implement:

configuring the node to receive information on a first method of the first object in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to perform at least one of getting or setting at least one property of the first object during execution of the graphical data flow program.

68. (Currently Amended) The non-transitory memory medium of claim 61,

wherein said configuring the node to receive information on a first object comprises configuring the node to receive information on a first class of the first object, wherein the first object is instantiated from the first class.

69. (Currently Amended) A non-transitory memory medium comprising program instructions for creating a graphical data flow program, wherein the program instructions are executable to implement:

displaying on a display a graphical data flow program, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

displaying on the display a node in the graphical data flow program in response to user input, wherein the node is configurable to invoke a method of a software object instantiated from any of a plurality of software classes;

connecting the node to one or more other nodes in the graphical data flow program in response to user input;

configuring the node to receive information on a first class in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to invoke a method of a first object instantiated from the first class during execution of the graphical data flow program.

70. (Currently Amended) The non-transitory memory medium of claim 69,

wherein the lines interconnecting the plurality of nodes indicate that data produced by one node is used by another node.

71. (Currently Amended) The non-transitory memory medium of claim 69,

wherein said configuring comprises connecting the information on the first class to an input of the node.

72. (Currently Amended) The non-transitory memory medium of claim 69, wherein said configuring comprises:

displaying on the screen a list of possible classes; and

selecting the first class from the list of possible classes in response to user input.

73. (Currently Amended) The non-transitory memory medium of claim 69, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input;

displaying on the screen a list of possible classes from the selected library; and

selecting the first class from the list of possible classes in response to user input.

74. (Currently Amended) The non-transitory memory medium of claim 69, wherein the program instructions are further executable to implement:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to invoke the method of the first object; and

executing said execution instructions, wherein the node invokes the method of the first object during said executing.

75. (Currently Amended) The non-transitory memory medium of claim 69, wherein the program instructions are further executable to implement:

configuring the node to receive information on a first method of the first class in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to invoke the first method of the first object instantiated from the first class during execution of the graphical data flow program.

76. (Currently Amended) A non-transitory memory medium comprising program instructions for creating a graphical data flow program, wherein the program instructions are executable to implement:

displaying on a display a graphical data flow program, wherein the graphical data flow program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical data flow program, wherein the plurality of interconnected nodes are connected by lines which represent flow of data among the nodes;

displaying on the display a node in the graphical data flow program in response to user input, wherein the node is configurable to perform at least one of getting or setting at least one property of a software object instantiated from any of a plurality of software classes;

connecting the node to one or more other nodes in the graphical data flow program in response to user input;

configuring the node to receive information on a first class in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to perform at least one of getting or setting at least one property of a first object instantiated from the first class during execution of the graphical data flow program.

77. (Currently Amended) The non-transitory memory medium of claim 76,

wherein the lines interconnecting the plurality of nodes indicate that data produced by one node is used by another node.

78. (Currently Amended) The non-transitory memory medium of claim 76,

wherein said configuring comprises connecting the information on the first class to an input of the node.

79. (Currently Amended) The non-transitory memory medium of claim 76, wherein said configuring comprises:

displaying on the screen a list of possible classes;

selecting the first class from the list of possible classes in response to user input.

80. (Currently Amended) The non-transitory memory medium of claim 76, wherein said configuring comprises:

displaying on the screen a list of libraries associated with one or more servers;

selecting a library from the list of libraries in response to user input;

displaying on the screen a list of possible classes from the selected library;

selecting the first class from the list of possible classes in response to user input.

81. (Currently Amended) The non-transitory memory medium of claim 76, wherein the program instructions are further executable to implement:

constructing execution instructions in response to the graphical data flow program, wherein the execution instructions are executable to perform at least one of getting or setting at least one property of the first object; and

executing said execution instructions, wherein the node performs at least one of getting or setting at least one property of the first object during said executing.

82. (Currently Amended) The non-transitory memory medium of claim 76, wherein the program instructions are further executable to implement:

configuring the node to receive information on a first property of the first class in response to user input;

wherein~~, during execution of the graphical data flow program,~~ the node is ~~operable~~ configured to perform at least one of getting or setting the first property of the first object instantiated from the first class during execution of the graphical data flow program.